

---

# Asistente basado en procesamiento del Lenguaje Natural para aprendizaje del lenguaje de programación Java (junio 2022)

**Julio Ponce Camacho, Ángel Maldonado García**

Tecnológico de estudios superiores de Jocotitlán, Jocotitlán, CP 50700, México.

**ABSTRACT** El tiempo es algo fundamental en cualquier sector, y en el caso de la programación tiene el mismo impacto, sin embargo, en el momento en que un estudiante empieza a aprender un lenguaje de programación, pueden surgir diversas dudas que se busquen solucionar a través de internet, pero esto puede ser complejo debido a la gran cantidad de sitios existentes en la red, confundiendo al usuario y provocando una gran inversión de tiempo que pudo haberse utilizado para la codificación del software. Por tal motivo, el objetivo de esta investigación es la creación de un asistente basado en el procesamiento del lenguaje Natural para la consulta de información, específicamente del lenguaje de programación Java, con el fin de hacer más eficiente el tiempo en la consulta de información ya que se considera que a partir de la implementación del motor de voz a texto y el posterior procesamiento del lenguaje natural, se realizara la búsqueda en la web obteniendo resultados relevantes con un 90% de precisión a la petición original recibida por el usuario. La metodología implementada para este software fue una Ágil llamada SCRUM, donde se definieron los requisitos generales como lo son los módulos de reconocimiento e identificación de voz, el módulo de búsqueda y la estructura interna de almacenamiento, para después realizar el diseño que es con base a una arquitectura modelo vista controlador basada en archivos, implementándose a través de tecnologías como PyQt5, Speech Recognition, Pytsx3, Beautiful Soap entre otras, desarrollado con el lenguaje de programación python con el paradigma de programación orientada a objetos obteniendo resultados de 92% de precisión en las búsquedas por conceptos y sintaxis realizadas.

**PALABRAS CLAVE** Asistente Virtual, código, Java, programación, python, reconocimiento de voz, síntesis de voz, Web Scraping.

## I. INTRODUCCIÓN

Los asistentes inteligentes son una tecnología que en la actualidad se ha vuelto muy popular, debido a las múltiples funciones con las que cuentan, y de algún modo permiten una interacción natural entre los dispositivos y los seres humanos.

La educación es un factor muy importante en el desarrollo de una persona, por ende, la relación entre la tecnología y la educación son dos aspectos que deberían de manejarse de manera conjunta para un buen desempeño en el aprendizaje de diversos temas.

El objetivo de desarrollar este proyecto de investigación, es el implementar un asistente basado en PLN (Procesamiento de lenguaje natural) dedicado al aprendizaje del lenguaje de programación Java, contando con la característica de orientar al usuario sobre el uso de este lenguaje a partir de fuentes confiables y validadas, fomentando las bases para un buen desarrollo en el aprendizaje a través del reconocimiento de voz y síntesis de voz por parte del asistente, permitiendo la

comunicación entre el software y el usuario a través de lenguaje natural.

Se considera que el tema de los asistentes inteligentes es de gran relevancia en la actualidad, y probablemente tengan un impacto aun mayor en un futuro, y el ser parte de ese desarrollo es lo que motiva la realización de esta investigación, con el fin de implementar la creación de este tipo de software.

### A. DEFINICIÓN DEL PROBLEMA

Dado el contexto de un estudiante que comienza en el estudio de un lenguaje de programación, pueden surgir diversas dificultades que dan como resultado un aprendizaje lento y una comprensión del lenguaje baja. Una de estas adversidades a las que se enfrenta un programador novato es la de desconocer la sintaxis utilizada en un lenguaje de programación de alto nivel, como lo es Java, debido a que existe gran cantidad de información que involucra la inversión de alta cantidad de tiempo para determinar cuáles son los datos que se adecúan a

lo que se está buscando, contando con una baja precisión en la obtención de los datos necesarios, debido a que en la actualidad la red está repleta de información, pero esto en lugar de tomarse como una ventaja se puede convertir en una desventaja debido a la gran cantidad de la misma, llegando a causar confusión al lector de la información, debido a que entre todos los sitios existentes se puede perder fácilmente entre tantas fuentes, dando como consecuencia una inversión de tiempo alta.

El tiempo es el recurso más valioso en la vida de una persona, por esto mismo es importante aprovecharlo lo más que se pueda. Un programador debe de saber aprovechar su tiempo correctamente para enfocarlo a la resolución de problemas, y es que cuando se realiza la codificación de un software, pueden surgir diversas situaciones en las que se requiere de una consulta en la red para poder seguir avanzando, lo cual puede tomar mucho tiempo en la búsqueda de la solución, y es que entre la gran cantidad de sitios que hay es tardado analizar la información de cada uno, y en el caso de un novato es aún peor.

Por tales motivos, se plantea el desarrollo de un asistente basado en PLN (Procesamiento de Lenguaje Natural) que sea capaz de filtrar la información presentada al usuario final mediante la extracción de un fragmento textual que contenga la información consultada, todo ello con base a la priorización de los sitios mejor valorados estableciéndolos como repositorios principales para ser la fuente de consulta, y a su vez, el asistente contara con la capacidad de almacenar códigos propios del usuario, con el propósito de que el estudiante pueda consultar los algoritmos que en algún momento desarrolló y puedan ser de utilidad en la creación de futuros programas.

## **B. HIPÓTESIS**

A partir de la implementación del motor de voz a texto y el posterior procesamiento del lenguaje natural; se realizará la búsqueda en la web obteniendo resultados relevantes con un 70% de precisión a la petición original recibida por el usuario.

## **C. OBJETIVO GENERAL**

Desarrollar un asistente basado en el procesamiento del lenguaje Natural para la consulta de información confiable del lenguaje de programación Java presentada al usuario final mediante un fragmento de texto.

## **D. OBJETIVOS ESPECIFICOS**

- Investigar los antecedentes de los asistentes basados en PLN
- Implementar un motor de conversión de voz a texto
- Implementar un sistema de identificación de palabras clave.
- Crear una modulo dedicado a la búsqueda de información a través de la web con la capacidad de filtrar fuentes oficiales o validadas.
- Evaluar la funcionalidad del asistente.

## **II. REVISIÓN DEL ESTADO DEL ARTE**

Los asistentes inteligentes son una tecnología que se puede implementar en muchos ámbitos, desde el desarrollo de un asistente con un funcionamiento particular, hasta asistentes con el propósito de atender necesidades generales de un usuario. Esta tecnología ha estado presente desde el año 1960, por lo cual han surgido diversos trabajos que han incrementado el desarrollo de esta.

Cada proyecto realizado tuvo diversos aportes al campo de los asistentes virtuales, mencionado algunos en el año 2017 artículo realizado por Arteaga Maza et al [1], presenta la Implementación de un asistente tutor basado en computación cognitiva para su uso en entornos visuales de aprendizaje, con el fin de mejorar los conocimientos de los estudiantes, resolviendo dudas. El proyecto contó con solo una interfaz desarrollada a través del framework Node.js, con el servicio de conversación “IMB Watson”, con peticiones por teclado, lo cual redujo un poco la eficiencia de consulta de la información, pero cumplió correctamente con base a la materia.

El campo de los asistentes no solo está en lo académico o la ayuda personal, también abarca más áreas, un ejemplo es el proyecto de asistente para el hogar en español creado por Carlos Díaz Fernández [2] en el 2017. Implementado en una Raspberry Pi 2, con la capacidad de realizar tareas que un usuario le solicite con el uso de su propia voz obteniendo una respuesta de la petición y ejecutando una acción como el encendido de luces o dispositivos. El proyecto se desarrolló a partir de una biblioteca de Python “Speech Recognition”, con el cual se logró hacer el reconocimiento de voz. La precisión de este asistente era del 88.66% según las pruebas de precisión al reconocer las órdenes, sin embargo, las peticiones que ejecutaba eran muy limitadas.

Han existido proyectos enfocados a un sector específico, donde los beneficiados son una pequeña región, como ejemplo es el asistente virtual tipo chatbot, desarrollado por el autor Luis Felipe García [3] desarrollado en 2018. En este trabajo de investigación para titulación, se desarrolló un asistente virtual tipo chatbot, teniendo la característica de recibir peticiones por el teclado, teniendo el propósito de hacer más rápida la recepción de una solicitud de PQR's (El Sistema de Peticiones, Quejas, Reclamos y Sugerencias) en la corporación San Isidro “Colegio Anglo Americano”. Esto a través de un módulo de administración web para la interacción del usuario, con la única deficiencia de la identificación de palabras limitada.

El reconocimiento de voz es una tecnología que permite agilizar la ejecución de diversas acciones, un ejemplo de esto es el trabajo de tesis “Sistema de Seguridad Por Reconocimiento de Voz” escrito por los autores Eyra Pérez & Fernando Martínez et al [4] en el año 2018. En este trabajo de titulación se desarrolló un sistema de seguridad con la característica del uso del reconocimiento de voz con el fin de que el sistema pueda interactuar de una manera más fluida y precisa, para poder tomar decisiones rápidas y confiables. El

diseño e implementación de este sistema, fue realizado MATLAB con algoritmos para obtener entrada de audio. Un factor importante para evitar errores fue estar en un ambiente de 70 dB (Decibeles), existiendo el problema de la falta de reconocimiento del habla al cambiar emociones al tener una tasa de error del 40%.

### III. MARCO TEÓRICO

Antes de iniciar con el desarrollo del proyecto, es importante el conocer diversos términos que se utilizan en la implementación del trabajo de investigación desarrollado.

#### A. ASISTENTE VIRTUAL

Son herramientas inteligentes que ayudan a los usuarios a buscar información en un conglomerado de recursos web. El despliegue natural de los mismos se realiza en las propias páginas web, donde permiten resolver las dudas de los usuarios formuladas en lenguaje natural usando técnicas de Inteligencia Artificial como lo es el Procesamiento del lenguaje Natural (PLN). [5]

#### B. PROCESAMIENTO DE LENGUAJE NATURAL

El procesamiento del lenguaje natural consiste en la habilidad de una máquina para procesar información comunicada mediante el uso del lenguaje natural. Se podría decir que el PLN consiste en usar una expresión natural que pueda tener comunicación con la computadora directamente, por medio escrito o comando de voz, facilitando las órdenes o peticiones con el lenguaje. [6]

#### C. RECONOCIMIENTO DE VOZ

El reconocimiento del habla permite a un ser humano comunicarse con un ordenador. A grandes rasgos, consiste en que el ordenador captura la señal de voz que emite una persona a través de un micrófono, convirtiéndola en información digital. [7].

#### D. SÍNTESIS DE VOZ

La Síntesis de Voz, también conocida como Conversión de Texto a Voz (CTV), consiste en dotar al sistema de la capacidad de convertir un texto dado en voz. La voz del ordenador puede generarse uniendo las grabaciones que se han hecho, ya sean de palabras enteras, o fonemas, pero siempre intentando que el sonido producido parezca lo más natural posible e inteligible, encadenando correctamente los sonidos dentro del discurso. [8]

#### E. FICHEROS

Un fichero es un conjunto de información relacionada, grabada en el sistema de almacenamiento secundario y a la que se hace referencia mediante un nombre. [9]

#### F. WEB SCRAPING

Web Scraping hace referencia a las técnicas que permiten recolectar u obtener datos desde la web a través de métodos

automáticos o manuales. Con web Scraping se puede extraer el contenido de un HTML de sitios web para filtrar la información y almacenarla para después hacer uso de los datos de interés. [9]

### IV. METODOLOGÍA

La metodología que se implementara para llevar a cabo este proyecto se basa en la metodología ágil de software, SCRUM, debido a que el campo de desarrollo de este proyecto requiere mucha programación, esta metodología es idónea, teniendo en cuenta su particularidad de implementación de un desarrollo dividido por “Sprint”, que equivale a una cierta cantidad de tiempo en el que plantea hacer la implementación de aspectos específicos. Las fases y flujo de trabajo de la metodología se pueden ver en la Figura 1.

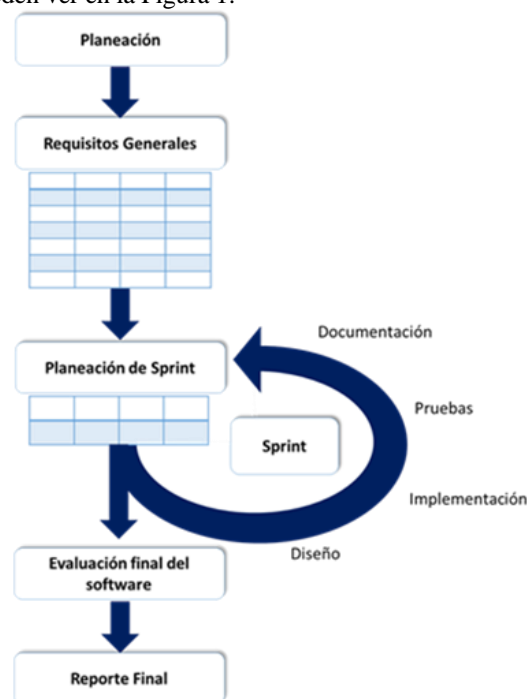


FIGURA 1. Fases de la metodología

#### 1) Planificación

En la etapa de planificación se desarrollarán diversas actividades que fungirán como guía para el inicio y culminación del proyecto, definiendo el tiempo inicio del desarrollo.

#### 2) Requisitos Generales

Los requisitos del software es un aspecto fundamental en esta metodología, ya que con base al producto que da como resultado esta fase, se podrá realizar la planificación de cada uno de los Sprint.

#### 3) Planeación de Sprint

Se definen los objetivos que se plantean lograr alcanzar en el sprint.

#### 4) Sprint

El Sprint es la etapa donde se realiza la implementación del proyecto, en el cual se realizan actividades importantes que incrementan el avance general del proyecto.

- **Diseño:** En esta etapa se diseña el funcionamiento lógico del software determinando como será el flujo de funcionamiento, el diseño técnico, lógico y visual.
- **Implementación:** Esta fase es donde se realiza la codificación del software a través de las tecnologías y algoritmos diseñados.
- **Pruebas:** Durante las Pruebas, se hace una evaluación de tipo unitaria de lo desarrollado en el Sprint.
- **Documentación:** Se documentan los avances generales del sprint.

##### 5) Evaluación Final del software

Ejecución de una serie de pruebas rigurosas al software con el fin de determinar si el objetivo planteado en la fase de planeación se logró alcanzar.

##### 6) Reporte Final

Se realiza un reporte general del proyecto realizado, redactando todas las funcionalidades del producto desarrollado, el proceso seguido desde su inicio y un análisis de los resultados obtenidos, para que al final el proyecto se dé por concluido.

#### A. PLANIFICACIÓN

El presente proyecto se planteó desarrollarse en un tiempo de 4 meses, donde la primera parte se contempla que sea el diseño y codificación del mismo, por lo que es la que hará uso de más tiempo a partir de la fecha 25 de marzo al 15 de junio del 2022, y la segunda parte será la evaluación del software para determinar si el proyecto realizado cumplió con los objetivos planteados al inicio del proyecto.

#### B. REQUERIMIENTOS GENERALES

Para el desarrollar cualquier tipo de proyecto, es importante determinar cuál va a ser el uso que este va a tener cuando se haya completado. En la figura 2 se presenta un diagrama con los diferentes casos de uso.

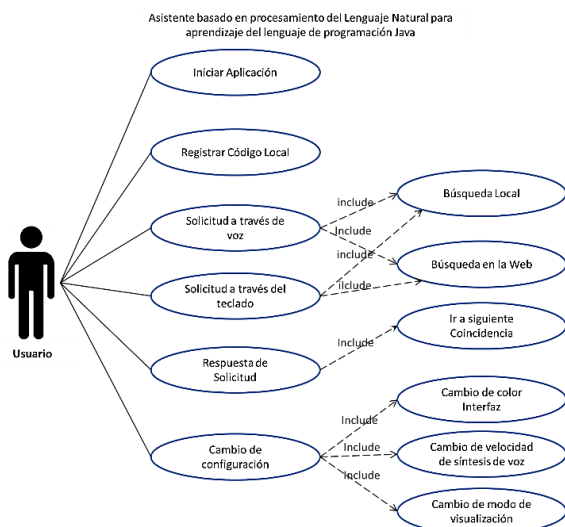


FIGURA 2. Diagrama de casos de uso

Dentro de un software no solo se puede presentar un solo tipo de uso, ya que entre las interfaces quizá se puedan realizar

diversos tipos de acciones con las que el usuario puede interactuar y es por ello que determinar los casos de uso antes del desarrollo es un requisito importante ya que con base a ese modelo se pueden determinar cada uno de los usos que va a tener el software y de esta manera se pueda desarrollar tal cual se plane.

En este caso conforme a la Figura 2 se puede apreciar que los casos de uso principales son el uso registro de código local, la búsqueda web y local por código o por concepto, así como la configuración de parámetros del software.

#### C. PLANIFICACIÓN DE SPRINT

El desarrollo como previamente se mencionó en la etapa de planeación, se consideró realizarse en 2 etapas, la primera que es la creación del asistente y la segunda es para realizar la evaluación de funcionamiento del software, por ende, las actividades a realizar se dividieron en diversos Sprint, los cuales en este caso corresponden a una semana, en la cual se desarrollaran diversas actividades de los requisitos generales

#### D. SPRINT

##### 1) Diseño

Con base a los requisitos que se determinaron previamente en fases anteriores de la metodología, se definió de manera precisa cual es el funcionamiento lógico del asistente, el cual cuenta con diversos módulos que permiten que cumpla con el propósito para el que se programó.

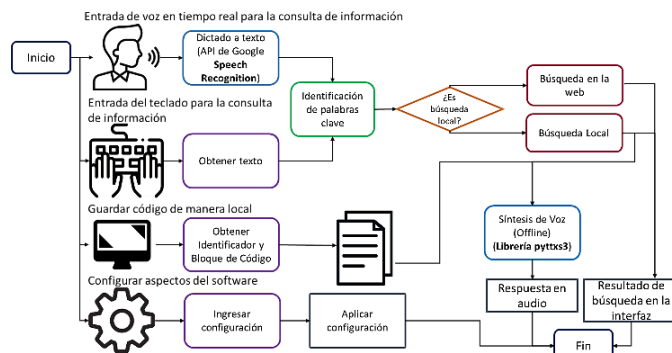


FIGURA 3. Diseño lógico del software

En la figura 3 se puede observar un diagrama que representa como se ejecuta el asistente, teniendo diversos caminos a seguir dependiendo lo que el usuario decida realizar. En total son 4 módulos el dictado por voz para realizar una consulta sea de tipo web o local, la consulta a través del teclado que comparte instrucciones con el módulo previo; el módulo de guardar código de manera local, en el cual con base a un identificado y un bloque de código se puede almacenar en el archivo de códigos locales para consultas rápidas, así como el módulo de configuración, en el cual simplemente el usuario determina parámetros del software de su preferencia y los aplica.

El modelo vista controlador, es un patrón de diseño que permite separar la base de datos (Modelo) la vista (GUI) y el controlador que permite la conexión entre los elementos previos. En este caso, el modelo son los archivos almacenados



localmente, los cuales se consultan constantemente con funciones programadas y se despliegan a la vista para que el usuario acceda a la información de manera clara. En la figura 4 se representa la distribución del asistente a través de este patrón de diseño.

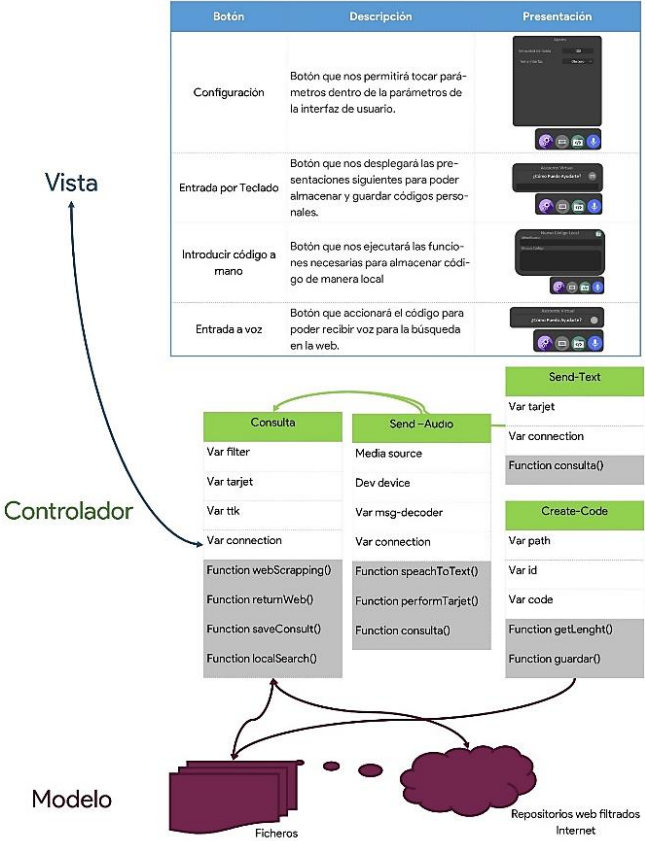


FIGURA 4. Modelo vista controlador

Los ficheros en este proyecto cumplen el rol de una base de datos, ya que en estos se almacenará información de utilidad para mantener un buen funcionamiento en cada uno de los módulos del software, ya que cada uno de ellos usa uno o más ficheros.

En la mayoría de ficheros a utilizar siguen un formato de lista, accediendo a través del nombre y obteniendo al valor que tenga enfrente, sin embargo, el fichero de códigos locales es diferente y sigue el formato que se puede ver en la figura 5, en el cual se explica gráficamente cómo funciona.

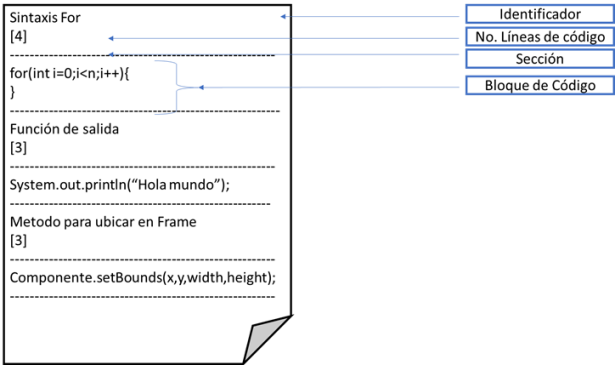


FIGURA 5. Formato de código local

Los códigos se almacenan en 4 partes, el identificador que es la parte superior, el número de líneas del bloque de código (Contando las secciones) el cual servirá para extraer solo el bloque de código o saltar a otro identificador en caso de que no sea lo que se busca.

2) Implementación

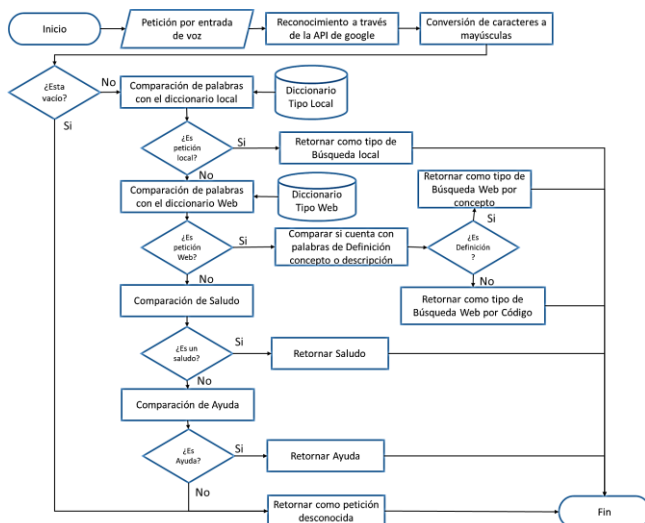
Python por defecto cuenta con gran cantidad de librerías y funciones que permiten desarrollar proyectos correctos y funcionales. En la tabla 1 se enlistan todas las dependencias necesarias para la codificación del proyecto más una previa descripción. Estas librerías tienen el propósito de desarrollar interfaces avanzadas, la implementación de la síntesis y reconocimiento de voz, así como la búsqueda por Web Scraping.

TABLA 1. Tecnologías implementadas con python

Paquete	Descripción
PyQt5	Es un set de módulos del framework Qt5 para la creación de interfaces graficas avanzadas
PipWin	Paquete que permite una instalación de paquetes con mayor compatibilidad para sistemas operativos Windows.
Speech Recognition	Librería que permite el reconocimiento de voz a traves de diversas API's como Google, Google Cloud, IBM, Sphinx etc.
Pytxs3	Paquete para python para la conversión de texto a voz de manera offline.
Pyaudio	Proporciona enlaces para hacer uso de los dispositivos de entrada y salida de audio.
PlaySound	Permite la reproducción de archivos de audio a través de los distintos dispositivos de audio.
Google	Permite la conexión con servicios de Google.
Google Search	Permite el acceso y uso del buscador de Google para realizar consultas en la red.
Beautiful Soap	Extrae información con base a etiquetas específicas del HTML de un sitio consultado.

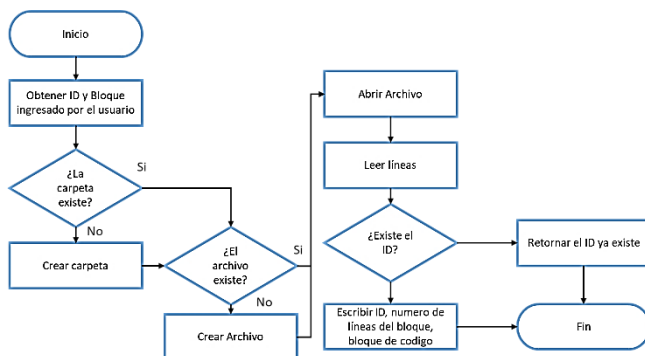
Una de las características de un asistente virtual es el uso del procesamiento del lenguaje natural, aceptando entradas a partir del teclado o por medio de la voz, en este caso se hizo énfasis en que se pudieran realizar las peticiones desde ambos medios.

En la figura 6 se puede observar el flujo que sigue la identificación de las peticiones que realiza un usuario, donde primero se realiza la entrada por voz por medio del micrófono, para después con base a Google Speech realizar la conversión de voz a texto. Una vez obtenido el texto se convierte en mayúsculas para evitar problemas al comparar las diferentes cadenas que permiten la identificación. Una vez obtenido el texto se verificar que realmente se haya reconocido, después se realiza la comparación con un diccionario de palabras para realizar una búsqueda local, si no hay coincidencia de un 80% se va a la comparación de palabras para búsqueda web, de lo contrario se verifica si es un saludo o una petición de ayuda, en caso de que alguno de estos si coincida devuelve una cadena para poder procesar la solicitud, limpiando las palabras que no sean de utilidad y realizar el siguiente proceso solo con los datos más importantes.



**FIGURA 6.** Diagrama de flujo del funcionamiento del reconocimiento de voz

Para ingresar localmente un código para poder tenerlo a disposición en todo momento se requiere de un identificador y un bloque de código el cual se almacenará en una carpeta y un archivo almacenado en el mismo directorio del ejecutable. En la figura 7 se puede ver el proceso que se realiza para añadir códigos a estos archivos, en los cuales principalmente se verifica si la carpeta y archivo existen, en caso contrario los crea para después analizar el archivo, cabe mencionar que no se realiza un análisis línea por línea, ya que el fichero local hace uso de un formato que permite saltar líneas para ganar tiempo con base al número de líneas de los códigos ya almacenados y únicamente leer el identificador del código para determinar si el identificador ya existe y evitar valores duplicados, en caso de que no, se añade al final del archivo.



**FIGURA 7.** Manejo de inserción de código local

Al consultar el archivo local para obtener algún código que el usuario necesita, este se desplegará en una ventana donde el bloque de código se puede editar, con el fin de que con base a dos botones el usuario pueda actualizar el código almacenado o bien eliminarlo.

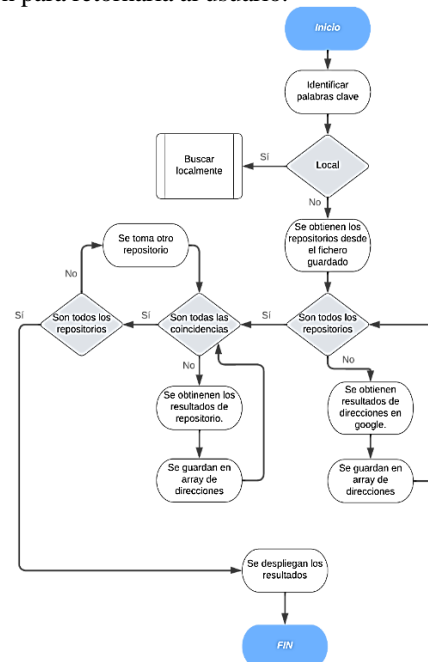
Para borrar los códigos almacenados localmente se sigue el siguiente proceso:

1. Inicio
2. Abrir el archivo de códigos locales.

3. Leer el número de líneas del archivo y almacenarlas en una variable
4. Abrir nuevamente el archivo y leer realizando saltos de líneas con base al número de líneas por bloques de código.
5. Si el ID a borrar coincide con el archivo analizado, almacenar el ID, el número de líneas y el bloque de código en una variable de tipo cadena.
6. Reemplazar el texto que coincida con la cadena anterior en la variable donde se almacenaron las líneas de todo el archivo.
7. Sobrescribir el archivo de código local con el resultado del reemplazo del texto.
8. Fin

Para actualizar un código local se ejecuta el algoritmo para borrarlo y después lo agrega nuevamente con la estructura actualizada.

Para la búsqueda de datos por internet se hizo uso de técnicas de web scraping a partir del buscador de Google y BeautifulSoup para obtener la información de las etiquetas del HTML. Este algoritmo se puede ver en la figura 8, en el cual primero se identifica el mensaje, para después identificar que sea una búsqueda de tipo web y con base a los repositorios que están en listados en ficheros, se realiza la búsqueda en aquellos sitios, tomando las dos primeras coincidencias y realizando la inspección para retornarla al usuario.



**FIGURA 8** Algoritmo de Web Scraping

La codificación de la interfaz gráfica se desarrolló con base al framework Qt5 a través de los módulos de la librería PyQt5, la cual permite el desarrollo de componentes gráficos avanzados, haciendo uso de tecnologías como CSS y HTML. Cada una de las ventanas cuenta con una barra de acciones integrada y propia para realizar el cerrado y minimizado, esto con el propósito de que la interfaz no sea afectada visualmente

cuando se tiene otro tema o una versión diferente del sistema operativo, evitando errores de visualización.

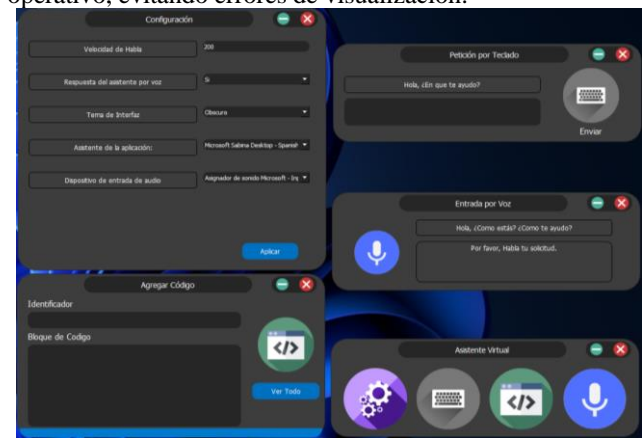


FIGURA 9 Interfaces del software desarrollado

### 3) Pruebas

Una vez desarrollado cada uno de los módulos necesarios y codificado cada una de las interfaces, el resultado final es el que se muestra en la figura 10, pudiendo realizar el procesamiento del lenguaje natural vía teclado o por medio de la voz para después realizar una consulta a través de la web con base a técnicas como web scraping o bien una consulta en el archivo local.

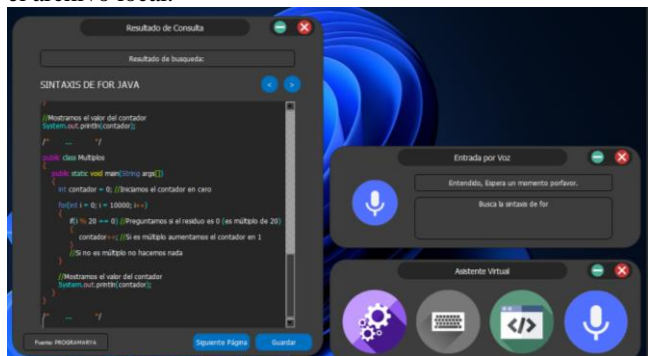


FIGURA 10 Funcionamiento del software

El consumo del software es bastante bajo para un hardware con 4 núcleos y 8 GB de RAM, consumiendo tan solo 40 MB de RAM y el 0.60% al procesar una petición del usuario, En la figura 11 se puede ver gráficos del consumo general del software.

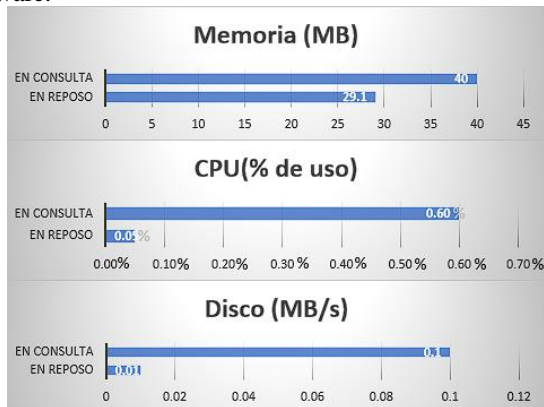


FIGURA 11 Consumo del software desarrollado

## V. ANALISIS DE RESULTADOS

El software desarrollado cumplió correctamente con las funciones de consulta planteadas, sin embargo, para comprobar si se logró verificar si la hipótesis se realizaron diversas pruebas con diferentes tipos de ambientes.

La primera prueba se realizó con un muestreo de 30 situaciones en las que se realizaron consultas por internet sobre conceptos y sintaxis referentes a la programación de java, y como se ve en la figura 11, en el 90% de estas se tuvieron resultados satisfactorios, con una tasa de error del 10%.



Figura 12 Pruebas de la precisión de búsqueda web

La consulta por internet tiene la precisión deseada, por lo cual se prosiguió con realizar pruebas de la detección de la voz con diversos parámetros, en la figura 12 se puede apreciar que en condiciones óptimas con poco ruido en el ambiente la precisión es del 93% en el reconocimiento, sin embargo, cuando estas condiciones malas con bastante ruido tienden a bajar a un 40% de precisión.

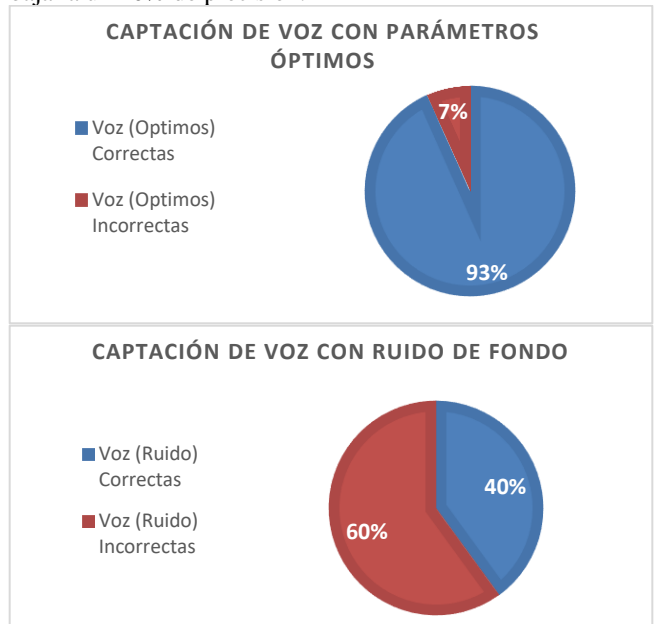
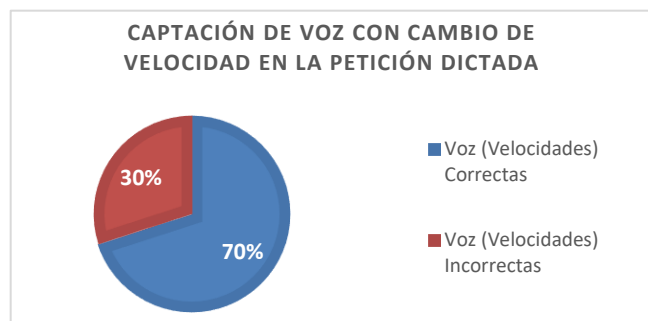


Figura 13 Pruebas del reconocimiento de voz con diferentes condiciones de ruido.

Un factor igual de relevante a la hora de implementar el procesamiento del lenguaje natural, es el reconocer el habla sin importar con qué tipo de emoción lo mencione o a qué velocidad realice la pronunciación de las palabras, por ende, se realizaron de igual forma pruebas con cambios en la

velocidad a la hora de que el usuario dicte una petición y como se ve en la figura 14, la precisión que logra a la hora de reconocer una petición a gran velocidad es del 70%, reduciendo hasta un 20% que cuando se presenta la consulta en condiciones normales.



**Figura 14** Pruebas de reconocimiento al cambiar la velocidad de la petición del hablante.

## VI. CONCLUSIÓN

La tecnología debe ser utilizada para el crecimiento del conocimiento de la humanidad, en este caso el asistente inteligente para el aprendizaje del lenguaje java cumplió con los objetivos, como la búsqueda en la web mediante reconocimiento del lenguaje natural y su procesamiento, de manera local con los ficheros implementados y sus respectivos identificadores, así como resultados más que aceptables en el conjunto general de funcionalidades implementadas.

El presente proyecto tuvo el objetivo principal de desarrollar un asistente que apoye a los estudiantes que desean aprender el lenguaje de programación java, por lo cual, este software estaba más enfocado en realizar la búsqueda de información por internet o localmente con una precisión del 90% en los resultados para que el usuario obtenga la información que le sea de ayuda, y con base a las pruebas realizadas este porcentaje de precisión se alcanzó adecuadamente, con un reconocimiento del habla del 93% en condiciones adecuadas.

Los puntos a mejorar de este proyecto son especialmente en la validación del reconocimiento en condiciones donde no sean favorables para el asistente ya que por ejemplo en la captación de voz con ruido de fondo la tasa de error aumento hasta en un 60% lo cual no es nada optimo. Otro aspecto a considerar es la implementación de algún motor de reconocimiento de voz offline en un futuro para que el asistente pueda entender la voz sin importar que no tenga conexión a internet el dispositivo, esto a través de tecnologías como por ejemplo Sphinx, siendo que sería de gran utilidad ya que el asistente puede realizar consultas de manera local con base al fichero que almacena esos datos.

Se puede concluir que el asistente virtual basado en procesamiento de lenguaje natura para el aprendizaje del lenguaje Java se realizó correctamente, sin embargo, hay detalles que pueden ser mejorables y que en futuras versiones se pueden corregir o bien escalar el proyecto a más lenguajes de programación ya que las bases de java se pudieron implementar de una manera correcta y funcional.

## REFERENCIAS

- [1] A. Maza, S. Daniel, "Análisis, diseño e Implementación de un asistente tutor basado en computación cognitiva para su uso en entornos visuales de aprendizaje", trabajo de titulación, Área técnica, Universidad técnica particular de la roja, 2017.
- [2] G. Juan, "Algoritmos y programación para docentes", Fundación Gabriel Piedrahita Uribe, Segunda Edición, Colombia, 2017.
- [3] G. Luis, "Asistente virtual tipo ChatBot." Trabajo de titulación, Universidad Católica de Colombia, Facultad de Ingeniería, Bogotá, 2018.
- [4] D. Carlos, "Diseño e implementación de un asistente personal inteligente en español basado en reconocimiento de voz empleando Raspberry Pi", Trabajo fin de grado, Universidad Carlos III de Madrid, 2018.
- [5] M. Damián, "Redes Neuronales: Conceptos básicos y aplicaciones", Reporte de investigación, Facultad regional rosario departamento de ingeniería química, pp. 8-11, 2001.
- [6] M. Dorys, C. Iván, G. Karolina, Q. Andrea, M. Christian, G. Teresa "Análisis del Estado Actual de Procesamiento de Lenguaje Natural", Revista Ibérica de Sistemas e Tecnologías de Información, RSTI, N° E42, pp. 126-136, 2020.
- [7] S. Adrián, "Puesta a punto de un robot real autónomo inteligente usando ROS", Trabajo de fin de grado, Universidad politécnica de Cartagena, ciudad de Cartagena Colombia, 2021.
- [8] C. Edwin, "Desarrollo de una aplicación de seguridad vial usando el Sistema de Posicionamiento Global (GPS); para prevenir accidentes vehiculares", Tesis profesional, Universidad peruana Unión, ciudad de Juliaca, Perú, 2020.
- [9] D. Glez, A. Lourencio & H. López, "Web scraping technologies in an API world", Artículo científico, Briefing in bioinformatics Vol. 15. pp 788-79, Abril 2013.



**Julio Ponce Camacho** nació en Ixtlahuaca estado de México, lugar en el cual creció y desarrollo. Actualmente cursa la universidad en el Tecnológico de estudios superiores de Jocotitlán para ser ingeniero en sistemas computacionales. Sus principales interés y metas están orientadas a desarrollarse como desarrollador de software y seguir aprendiendo acerca del mundo de la tecnología.



**Ángel Maldonado García** nació en Ixtlahuaca, estado de México. Actualmente es estudiante del 8vo semestre de ingeniería en sistemas computacionales en el tecnológico de estudios superiores de Jocotitlán, cuenta con conocimientos medios en patrones de diseño de software, enfocado a convertirse en desarrollador full stack de aplicaciones web y móviles